Product Formulas for Exponentials of Commutators

Andrew M. Childs and Nathan Wiebe

Department of Combinatorics & Optimization, University of Waterloo, Ontario N2L 3G1, Canada and
Institute for Quantum Computing, University of Waterloo, Ontario N2L 3G1, Canada

We provide a recursive method for constructing product formula approximations to exponentials of commutators, giving the first approximations that are accurate to arbitrarily high order. Using these formulas, we show how to approximate unitary exponentials of (possibly nested) commutators using exponentials of the elementary operators, and we upper bound the number of elementary exponentials needed to implement the desired operation within a given error tolerance. By presenting an algorithm for quantum search using evolution according to a commutator, we show that the scaling of the number of exponentials in our product formulas with the evolution time is nearly optimal. Finally, we discuss applications of our product formulas to quantum control and to implementing anticommutators, providing new methods for simulating many-body interaction Hamiltonians.

I. INTRODUCTION

Product formulas provide a way of approximating a single operator exponential with a product of simpler operator exponentials. Such formulas are useful in numerical analysis, where they can be applied to the solution of differential equations (see for example [1]). More recently, product formulas have become a key tool for quantum simulation [2–10]. Hamiltonian simulation using product formulas has numerous applications in quantum information processing, including simulating quantum mechanics [2], implementing continuous-time quantum algorithms [4, 11–13], and quantum control techniques (see for example [14]).

The primary application of product formulas is to represent exponentials of sums. Although exponentials of commutators are not as ubiquitous, they arise naturally via their role in Lie groups. Exponentials of commutators appear in numerous asymptotic expansions, including the Baker–Campbell–Hausdorff series and the Magnus expansion. They also play a role in quantum computation, such as in the Solovay–Kitaev theorem [15], which constructively proves that any finite universal gate set is sufficient to perform efficient universal quantum computation, and in quantum control [14], where product formulas for exponentials of commutators can be used to suppress couplings or introduce ones that are not naturally present.

Although the theory of product formula approximations for exponentials of sums is well understood, it is considerably less developed in the case of commutator exponentials. Product formula approximations to commutator exponentials approximate an exponential of the form $\exp([A, B]T)$ for operators A and B and a real number T with a sequence containing exponentials of A and of B. In the limit of small T, low-order product formulas for exponentials of commutators are well known, but we are not aware of any previous systematic method of constructing high-order approximations.

In this work, we construct arbitrarily high-order product formula approximations to exponentials of commutators. Our formulas are analogous to Suzuki's seminal work on product formulas [16], but apply to the case where the exponentiated operator is a commutator, rather than a sum, of two operators. We do not explicitly consider cases where the operator is a sum of commutators or is an ordered commutator exponential, but such cases can be addressed by combining our results with existing product formula approximations for exponentials of sums [16] (see [8] for an improved analysis in some cases) or ordered operator exponentials [7, 9, 17].

Our results provide a method to simulate exponentials of the form $e^{[A,B]T}$, for any desired $T \in \mathbb{R}$, using devices that can enact evolution under A or B separately. Specifically, we imagine that we have a pair of devices $\mathcal{D}_A$ and $\mathcal{D}_B$ that take as input an evolution time t and perform operations e^{At} and e^{Bt}, respectively. Physically, we can imagine that $\mathcal{D}_A$ and $\mathcal{D}_B$ represent control fields that enact a desired evolution in a quantum system and t represents the time for which those control fields are applied. Alternatively, we can imagine that iA and iB are Hamiltonians that can be easily simulated and that $\mathcal{D}_A$ and $\mathcal{D}_B$ represent quantum simulation algorithms performing the corresponding evolution. In either case, we measure the efficiency of our formulas by the number of times these devices need to be used to simulate $e^{[A,B]T}$. Note that our figure of merit is *not* the total amount of time the control fields are applied: an elementary evolution e^{At} or e^{Bt} has unit cost independent of t.

The remainder of the paper is organized as follows. In Section II we present basic product formulas for approximating exponentials of commutators. We combine these formulas to approximate exponentials of nested commutators in Section III. Upper bounds for the approximation errors are derived in Section IV. These error bounds are applied in Section V, where we show that the resulting formulas use a number of exponentials that is only slightly superlinear in the evolution time. We show that this performance is nearly optimal in Section VI by proving that sublinear simulation would violate the quantum lower bound on the query complexity of unstructured search. We present

applications of our techniques in Section VII, including simple examples of quantum control as well as a method for simulating exponentials of anticommutators of operators. In particular, the simulation of anticommutators provides a novel method for implementing many-body interactions in quantum systems. We conclude in Section VIII with a discussion of the results and some open problems.

II. BASIC PRODUCT FORMULAS

In this section we present basic formulas approximating the operator $e^{[A,B]t^{k+1}}$ for small t as a product of powers of e^{At} and e^{Bt^k}, for a given positive integer k. Choosing $k = 1$ yields the most efficient formulas for the case where B can be implemented directly. We consider higher values of k for the case where B is itself a (possibly nested) commutator, as discussed further in Section III. We present two recursive constructions, one for k odd and another for k even, giving high-order approximation formulas for $e^{[A,B]t^{k+1}}$ in terms of exponentials of A and B. For every integer $p \geq 1$, we present a formula with approximation error $O(t^{2p+k+1})$ in the limit of small t.

Since the $k = 1$ case is the most natural, we begin with splitting formulas for the case where k is odd. We then discuss the simpler case where k is even in Section II B.

A. Odd-k Formulas

We now develop a recursive approximation-building method that can be used to construct an arbitrarily high-order approximation to $e^{[A,B]t^{k+1}}$ in terms of a product of powers of e^{At} and e^{Bt^k}, where k is odd. The construction uses an initial approximation to the time evolution, as follows.

Lemma 1. *Let A and B be bounded operators, let $k \geq 1$ be a real number, and define*

$$V_{1,k}(At, Bt^k) := e^{At}e^{Bt^k}e^{-At}e^{-Bt^k}. \tag{1}$$

Then $V_{1,k}(At, Bt^k) = e^{[A,B]t^{k+1}+O(t^{k+2})}$.

Proof. The Baker–Campbell–Hausdorff (BCH) formula implies

$$V_{1,k}(At, Bt^k) = e^{At+Bt^k+\frac{1}{2}[A,B]t^{k+1}+O(t^{k+2})}e^{-At-Bt^k+\frac{1}{2}[A,B]t^{k+1}+O(t^{k+2})}. \tag{2}$$

A second use of the BCH formula gives

$$V_{1,k}(At, Bt^k) = e^{[A,B]t^{k+1}+O(t^{k+2})}, \tag{3}$$

which completes the proof. $\square$

The product formula $V_{1,1}(At, Bt) = e^{At}e^{Bt}e^{-At}e^{-Bt}$ is known as the group commutator. This formula has many applications, including generating optimal control sequences [14] and approximating unitary gates via the Solovay–Kitaev theorem [15]. We show that higher-order generalizations of this formula can be constructed using an iterative approximation-building method that is reminiscent of Suzuki's method [16].

To use our technique, we must have product formulas for the inverses of our approximations. Our approximations built from $V_{1,k}(At, Bt^k)$ possess one of two symmetry properties that make their inverses simple to compute.

Definition 1. *A product formula U is* symmetric *if $U(X,Y) = U(Y,X)^{-1}$ and is* antisymmetric *if $U(X,Y) = U(-Y,-X)^{-1}$, for all bounded operators X and Y.*

In particular, $V_{1,k}$ is symmetric, and we will see that high-order approximations constructed using this product formula are either symmetric or antisymmetric.

Now we are ready to describe the main result of this section, which shows how to construct an arbitrarily high-order approximation.

Theorem 2. *Let A and B be bounded operators, let $k \geq 1$ be an odd integer, and let $V_{p,k}(At, Bt^k)$ be a product formula with $V_{p,k}(At, Bt^k) = e^{[A,B]t^{k+1}+O(t^{2p+k})}$ for some positive integer p. Let*

$$
\begin{aligned}
V_{p+1,k}(At, Bt^k) := {} & V_{p,k}(A\gamma_p t, B(\gamma_p t)^k)V_{p,k}(-A\gamma_p t, -B(\gamma_p t)^k) \\
& \times V_{p,k}(A\beta_p t, B(\beta_p t)^k)^{-1}V_{p,k}(-A\beta_p t, -B(\beta_p t)^k)^{-1} \\
& \times V_{p,k}(A\gamma_p t, B(\gamma_p t)^k)V_{p,k}(-A\gamma_p t, -B(\gamma_p t)^k)
\end{aligned}
\tag{4}
$$

where

$$\beta_p := (2r_p)^{1/(k+1)}, \qquad \gamma_p := (1/4 + r_p)^{1/(k+1)}, \qquad r_p := \frac{2^{\frac{k+1}{2p+k+1}}}{4\left(2 - 2^{\frac{k+1}{2p+k+1}}\right)}. \tag{5}$$

Then $V_{p+1,1}(At, Bt) = \exp([A,B]t^2 + O(t^{2(p+1)+1}))$ and $V_{p+1,k}(At, Bt^k) = \exp([A,B]t^{k+1} + O(t^{2(p+1)+k+1}))$ if $k > 1$. Furthermore, $V_{p+1,k}$ is symmetric if $V_{p,k}$ is antisymmetric and is antisymmetric if $V_{p,k}$ is symmetric.

Proof. The assumption that $\|U(t) - V_{p,k}(At, Bt^k)\| \in O(t^{2p+k})$, where $U(t) := e^{[A,B]t^{k+1}}$, implies that there exist operators $C(A,B)$, $D(A,B)$, and $E(A,B)$ such that

$$V_{p,k}(At, Bt^k) = U(t) + C(A,B)t^{2p+k} + D(A,B)t^{2p+k+1} + E(A,B)t^{2p+k+2} + O(t^{2p+k+3}). \tag{6}$$

Every term in the Taylor series of $V_{p,k}(At, Bt^k)$ is a product of powers of At and Bt^k, each of which contributes an odd power of t. Thus each term in $C(A,B)$ and $E(A,B)$ contains an odd total number of A and B operators. Similarly, because $2p+k+1$ is even, each term in $D(A,B)$ contains an even number of A and B operators. Therefore,

$$\begin{aligned}
C(A,B) &= -C(-A,-B)\\
D(A,B) &= D(-A,-B)\\
E(A,B) &= -E(-A,-B).
\end{aligned} \tag{7}$$

We use these properties to simplify $V_{p,k}(At, Bt^k)V_{p,k}(-At, -Bt^k)$ for arbitrary t. Specifically, we expand $V_{p,k}$ as a power series and use (6) to show that

$$\begin{aligned}
V_{p,k}(At, Bt^k)V_{p,k}(-At, -Bt^k) = U(t)^2 &+ [C(A,B)U(t) - U(t)C(A,B)]t^{2p+k}\\
&+ [D(A,B)U(t) + U(t)D(A,B)]t^{2p+k+1}\\
&+ [E(A,B)U(t) - U(t)E(A,B)]t^{2p+k+2} + O(t^{2p+k+3}).
\end{aligned} \tag{8}$$

We then Taylor expand each $U(t)$ (but not $U(t)^2 = U(2^{1/(1+k)}t)$) in (8) to lowest order in t and find that

$$V_{p,k}(At, Bt^k)V_{p,k}(-At, -Bt^k) = U(t)^2 + 2D(A,B)t^{2p+k+1} + O(t^{2p+k+3}) + O(t^{2p+2k+1}). \tag{9}$$

This implies that (4) has no error terms of order t^{2p+k} because it is a product of three pairs of product formula approximations of the same form as (9).

Next we show that the careful choice of β_p and γ_p eliminates the term of order t^{2p+k+1}. To do so, we relate the error terms of $V_{p,k}(At, Bt^k)$ and $V_{p,k}(At, Bt^k)^{-1}$. Similarly to (6), we have

$$V_{p,k}(At, Bt^k)^{-1} = U(t)^{-1} + \tilde{C}(A,B)t^{2p+k} + \tilde{D}(A,B)t^{2p+k+1} + \tilde{E}(A,B)t^{2p+k+2} + O(t^{2p+k+3}) \tag{10}$$

for some operators $\tilde{C}(A,B)$, $\tilde{D}(A,B)$, and $\tilde{E}(A,B)$. This expansion directly follows from the symmetry properties of $V_{p,k}$. For example, if $V_{p,k}$ is symmetric, then $V_{p,k}(At, Bt^k)^{-1} = V_{p,k}(Bt^k, At)$. We know from the previous discussion that $e^{[B,A]t^{k+1}} = V_{p,k}(Bt^k, At) + O(t^{2p+k+1}) = U(t)^{-1}$. The symmetry of the formula then implies that $U(t)^{-1} = V_{p,k}(At, Bt^k)^{-1} + O(t^{2p+k+1})$, which justifies (10). The anti-symmetric case follows similarly.

Since each term in $\tilde{C}(A,B)$ and $\tilde{E}(A,B)$ consists of an odd number of A and B operators and each term in $\tilde{D}(A,B)$ consists of an even number of such operators, we have (similarly to (7))

$$\begin{aligned}
\tilde{C}(A,B) &= -\tilde{C}(-A,-B)\\
\tilde{D}(A,B) &= \tilde{D}(-A,-B)\\
\tilde{E}(A,B) &= -\tilde{E}(-A,-B).
\end{aligned} \tag{11}$$

Equations (4), (9), and (10) then imply that

$$\begin{aligned}
V_{p+1,k}(At, Bt^k) &= \left(U(\gamma_p t)^2 + 2D(A,B)(\gamma_p t)^{2p+k+1}\right)\\
&\quad \times \left(U(\beta_p t)^{-2} + 2\tilde{D}(A,B)(\beta_p t)^{2p+k+1}\right)\\
&\quad \times \left(U(\gamma_p t)^2 + 2D(A,B)(\gamma_p t)^{2p+k+1}\right) + O(t^{\min\{2p+k+3, 2p+2k+1\}})\\
&= U(t) + \left[4\gamma_p^{2p+k+1}D(A,B) + 2\beta_p^{2p+k+1}\tilde{D}(A,B)\right]t^{2p+k+1} + O(t^{\min\{2p+k+3, 2p+2k+1\}}). \tag{12}
\end{aligned}$$

We can relate $D(A,B)$ to $\tilde{D}(A,B)$ by noting that

$$
\begin{aligned}
\mathbb{1} &= V_{p,k}(At, Bt^k)V_{p,k}(At, Bt^k)^{-1} \\
&= (U(t) + C(A,B)t^{2p+k} + D(A,B)t^{2p+k+1} + E(A,B)t^{2p+k+2}) \\
&\quad \times (U(t)^{-1} + \tilde{C}(A,B)t^{2p+k} + \tilde{D}(A,B)t^{2p+k+1} + \tilde{E}(A,B)t^{2p+k+2}) + O(t^{\min\{2p+k+3, 2p+2k+1\}}).
\end{aligned}
\tag{13}
$$

By Taylor expanding the resulting formula, we find $\tilde{D}(A,B) = -D(A,B)$ (as well as $\tilde{C}(A,B) = -C(A,B)$, and if $k > 1$ then $\tilde{E}(A,B) = -E(A,B)$).

Recalling the definitions of β_p and γ_p, we can substitute $D(A,B) = -\tilde{D}(A,B)$ into (12) to find

$$
V_{p+1,k}(At, Bt^k) = U(t) + \left[4(1/4 + r_p)^{\frac{2p+k+1}{k+1}} - 2(2r_p)^{\frac{2p+k+1}{k+1}}\right] D(A,B)t^{2p+k+1} + O(t^{\min\{2p+k+3, 2p+2k+1\}}). \tag{14}
$$

The value of r_p specified in (5) is a root of the expression in square brackets, so the term of order t^{2p+k+1} vanishes. This demonstrates that

$$
V_{p+1,k}(At, Bt^k) = e^{[A,B]t^{k+1}} + O(t^{\min\{2p+k+3, 2p+2k+1\}}) \tag{15}
$$

as claimed.

Finally, we must show that $V_{p+1,k}$ is either symmetric or antisymmetric. First, suppose $V_{p,k}$ is symmetric. Then

$$
\begin{aligned}
&V_{p+1,k}(At, Bt^k)V_{p+1,k}(-Bt^k, -At) \\
&= \Big(V_{p,k}(A\gamma_p t, B(\gamma_p t)^k)V_{p,k}(-A\gamma_p t, -B(\gamma_p t)^k)V_{p,k}(B(\beta_p t)^k, A\beta_p t) \\
&\quad \times V_{p,k}(-B(\beta_p t)^k, -A\beta_p t)V_{p,k}(A\gamma_p t, B(\gamma_p t)^k)V_{p,k}(-A\gamma_p t, -B(\gamma_p t)^k)\Big) \\
&\quad \times \Big(V_{p,k}(-B(\gamma_p t)^k, -A\gamma_p t)V_{p,k}(B(\gamma_p t)^k, A\gamma_p t)V_{p,k}(-A\beta_p t, -B(\beta_p t)^k) \\
&\quad \times V_{p,k}(A\beta_p t, B(\beta_p t)^k)V_{p,k}(-B(\gamma_p t)^k, -A\gamma_p t)V_{p,k}(B(\gamma_p t)^k, A\gamma_p t)\Big) = \mathbb{1}.
\end{aligned}
\tag{16}
$$

Thus $V_{p+1,k}$ is antisymmetric. A similar calculation shows that if $V_{p,k}$ is antisymmetric then $V_{p+1,k}$ is symmetric. $\qquad\square$

The result of Theorem 2 shows how to recursively construct the product formula $V_{p,k}$ starting from $V_{1,k}$. Note that this construction involves terms of the form $V_{q,k}(A\lambda t, B(\lambda t)^k)^{-1}$. Since $V_{q,k}$ is either symmetric or antisymmetric, its inverse can be represented explicitly using Definition 1.

The following corollary shows that symmetrization alone can increase the order of the approximation of $V_{p,1}$ from $O(t^{2p+1})$ to $O(t^{2p+2})$ at the cost of doubling the number of exponentials.

Corollary 3. *Let A and B be bounded operators, let $V_{p,1}(At, Bt)$ satisfy $V_{p,1}(At, Bt) = e^{[A,B]t^2 + O(t^{2p+1})}$, and define*

$$
V'_{p,1}(At, Bt) := V_{p,1}(At/\sqrt{2}, Bt/\sqrt{2})V_{p,1}(-At/\sqrt{2}, -Bt/\sqrt{2}). \tag{17}
$$

Then

$$
V'_{p,1}(At, Bt) = e^{[A,B]t^2 + O(t^{2p+2})}. \tag{18}
$$

Proof. This is a simple consequence of (9) with $k = 1$. Since every term of order $2p+1$ in $V_{p,1}(At, Bt)$ must contain an odd total number of As and Bs, the terms of order $2p+1$ in $V_{p,1}(At, Bt)$ and $V_{p,1}(-At, -Bt)$ are equal and opposite. By Taylor expanding $V_{p,1}(At/\sqrt{2}, Bt/\sqrt{2})V_{p,1}(-At/\sqrt{2}, -Bt/\sqrt{2})$, it is easy to see that the error terms of order $2p+1$ cancel. The error in $V'_{p,1}(At, Bt)$ is therefore $O(t^{2p+2})$ as claimed. $\qquad\square$

Note that a similar symmetrization would improve the error bound of $V_{p,k}$ from $O(t^{2p+k})$ to $O(t^{2p+k+1})$, but this gives no improvement over Theorem 2 for $k > 1$ since that theorem already shows that the error is $O(t^{2p+k+1})$.

For $k > 1$, we can apply Theorem 2 recursively to produce a formula $V_{p,k}$ with $4 \times 6^{p-1}$ exponentials having error $O(t^{2p+k+1})$. For $k = 1$, applying Theorem 2 recursively followed by one application of Corollary 3 gives a formula $V'_{p,1}$ with $8 \times 6^{p-1}$ exponentials having error $O(t^{2p+2})$.

Figure 1 presents a numerical example showing improved error scaling as p increases. This example considers simulating the commutator of the Pauli operators σ_x and σ_z using $k = 1$. These data suggest that our upper bound on the error scaling is in fact tight.
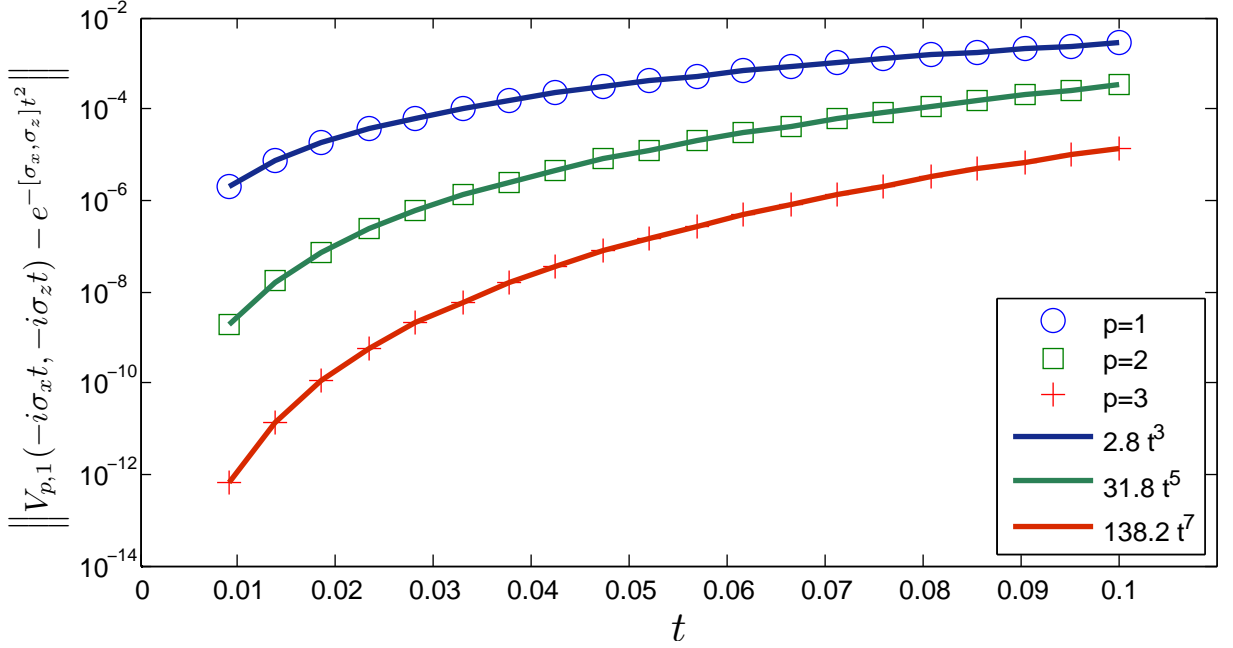
FIG. 1. Error scaling of the formulas $V_{p,1}(-i\sigma_x t, -i\sigma_z t)$ for $p = 1, 2, 3$ as approximations of $e^{-[\sigma_x,\sigma_z]t^2}$, where $\|\cdot\|$ is the 2-norm.

B. Even-k Formulas

If k is even then the above approach does not apply, so different reasoning must be used to generate high-order approximations to $e^{[A,B]t^k}$. We show, somewhat surprisingly, that Suzuki's recursive approximation-building formula [16] for the exponential of a sum can be used to generate arbitrarily high-order approximation formulas for commutators from a relatively simple initial approximation. The initial formula is as follows.

Lemma 4. *Let A and B be bounded operators, let $\xi_k := 2^{-1/(1+k)}$, let $k > 0$ be even, and define*

$$W_{1,k}(At, Bt^k) := e^{At\xi_k} e^{Bt^k \xi_k^k} e^{-2At\xi_k} e^{-Bt^k \xi_k^k} e^{At\xi_k}. \tag{19}$$

Then

$$e^{[A,B]t^{k+1}} = W_{1,k}(At, Bt^k) + O(t^{k+3}). \tag{20}$$

Proof. Since $U(t) := e^{[A,B]t^{k+1}} = e^{[B,-A]t^{k+1}}$, it follows from Lemma 1 (which applies for any k) that

$$e^{[A,B]t^{k+1}} = V_{1,k}(At\xi_k, Bt^k\xi_k^k) V_{1,k}(Bt^k\xi_k^k, -At\xi_k) + O(t^{k+2})$$

$$= \left(e^{At\xi_k} e^{Bt^k\xi_k^k} e^{-At\xi_k} e^{-Bt^k\xi_k^k}\right) \left(e^{Bt^k\xi_k^k} e^{-At\xi_k} e^{-Bt^k\xi_k^k} e^{At\xi_k}\right) + O(t^{k+2}). \tag{21}$$

The dominant term of equation (21) reduces to $W_{1,k}(At, Bt^k)$ by simplifying the middle terms.

It remains to show that the term proportional to t^{k+2} vanishes. This follows from the fact that $W_{1,k}(At, Bt^k) = W_{1,k}(-At, Bt^k)^{-1}$. Thus, defining $C(A, B)$ via $W_{1,k}(At, Bt^k) = U(t) + C(A,B)t^{k+2} + O(t^{k+3})$, we have

$$\mathbb{1} = W_{1,k}(At, Bt^k)W_{1,k}(-At, Bt^k) = \mathbb{1} + (U(t)C(-A,B) + C(A,B)U(-t))t^{k+2} + O(t^{k+3}). \tag{22}$$

Because k is even, C must be composed of an even number of As, so $C(-A, B) = C(A, B)$. Since (22) must hold for arbitrary t, we have $C(A, B) = 0$. □

The product formula in Lemma 4 is reminiscent of the Strang splitting formula, and indeed reduces to that approximation to $\mathbb{1} = e^{(A-A)t}$ when $B = \mathbb{1}$, or if we take $B = \mathcal{B}^k$ for some operator $\mathcal{B}$ and substitute $k = 0$ into the resulting formulas. Since Suzuki's iterative approximation-building method can refine the Strang splitting into arbitrarily high-order formulas, one might suppose that the same recursion could approximate exponentials of commutators. We make this intuition precise in the following theorem.

Theorem 5. *Let A and B be bounded operators, let $k > 0$ be an even integer, let $p \geq 1$ be an integer. Suppose that $W_{p,k}(At, Bt^k) = W_{p,k}(-At, Bt^k)^{-1}$ and $W_{p,k}(At, Bt^k) = e^{[A,B]t^{k+1} + O(t^{2p+k+1})}$, and define*

$$W_{p+1,k}(At, Bt^k) := W_{p,k}(A\nu_p t, B(\nu_p t)^k)^2 W_{p,k}(-A\mu_p t, B(\mu_p t)^k) W_{p,k}(A\nu_p t, B(\nu_p t)^k)^2 \tag{23}$$

where

$$\mu_p := (4s_p)^{1/(k+1)}, \qquad \nu_p := (1/4 + s_p)^{1/(k+1)}, \qquad s_p := \frac{4^{\frac{k+1}{2p+k+1}}}{4\left(4 - 4^{\frac{k+1}{2p+k+1}}\right)}. \tag{24}$$

Then

$$W_{p+1,k}(At, Bt^k) = \exp\left([A, B]t^{k+1} + O(t^{2p+k+3})\right). \tag{25}$$

Proof. We follow the same reasoning used to analyze the Suzuki formulas. We have

$$W_{p,k}(At, Bt^k) = U(t) + C(A, B)t^{k+2p+1} + O(t^{2p+k+2}) \tag{26}$$

for some $C(A, B)$. Substituting (26) into (23) gives

$$W_{p+1,k}(At, Bt^k) = U(t) + \left(4C(A, B)\nu_p^{2p+k+1} + C(-A, B)\mu_p^{2p+k+1}\right)t^{2p+k+1} + O(t^{2p+k+2}). \tag{27}$$

Each term comprising $C(A, B)$ must contain an odd number of As, because each A is associated with t and each B is associated with t^k, so since k is even, a term proportional to t^{2p+k+1} can only be formed from an odd number of As. Thus $C(A, B) = -C(-A, B)$. Therefore, the coefficient of $C(A, B)$ is zero when

$$4(1/4 + s_p)^{(2p+k+1)/(k+1)} - (4s_p)^{(2p+k+1)/(k+1)} = 0. \tag{28}$$

The value of s_p specified in (24) is a root of (28) and hence

$$\|U(t) - W_{p+1,k}(At, Bt^k)\| \in O(t^{2p+k+2}). \tag{29}$$

Finally, we show that by symmetry properties of $W_{p+1,k}$, the actual error scaling is better than in (29). We have

$$W_{p+1,k}(At, Bt^k) = U(t) + D(A, B)t^{2p+k+2} + O(t^{2p+k+3}). \tag{30}$$

Since $W_{p,k}(At, Bt^k) = W_{p,k}(-At, Bt^k)^{-1}$, it is easy to see by multiplication and (23) that $W_{p+1,k}(At, Bt^k) = W_{p+1,k}(-At, Bt^k)^{-1}$. We conclude that $D(A, B) = 0$ by the same argument used in (22). Since $D(A, B) = 0$, we have $W_{p+1,k}(At, Bt^k) = U(t) + O(t^{2(p+1)+k+1})$ as claimed. $\square$

The output from the recursive formula in Theorem 5 can be used as input, so the theorem shows how to refine $W_{1,k}$ into an arbitrarily high-order approximation. The basic formula $W_{1,k}$ has 5 exponentials, and each iteration increases the number of exponentials by a factor of 5, so $W_{p,k}(At, Bt^k)$ consists of 5^p exponentials.

Figure 2 presents a numerical example showing improved error scaling as p increases with $k = 2$. As in the case of odd k, these data suggest that our upper bound on the error is tight.

III. EXPONENTIALS OF NESTED COMMUTATORS

In general, we may be interested in approximating exponentials of nested commutators of the form

$$Z_k := [A_k, [A_{k-1}, [\ldots, [A_1, A_0]\ldots]]] \tag{31}$$

(i.e., $Z_0 = A_0$ and $Z_k = [A_k, Z_{k-1}]$ for $k > 0$). Our strategy for approximating such exponentials is simple. For example, suppose k is odd. We first approximate $e^{Z_k t^{k+1}}$ with $V_{p,k}(A_k t, Z_{k-1} t^k)$. The resulting expression still contains commutator exponentials because of the presence of exponentials of Z_{k-1}. We approximate each exponential of the form $e^{Z_{k-1}(\lambda t)^k}$ by $W_{p,k-1}(A_{k-1}\lambda t, Z_{k-2}(\lambda t)^{k-1})$, leaving exponentials of Z_{k-2} in the formula. This process is repeated recursively until no exponentials of commutators remain. The case where k is even can be addressed
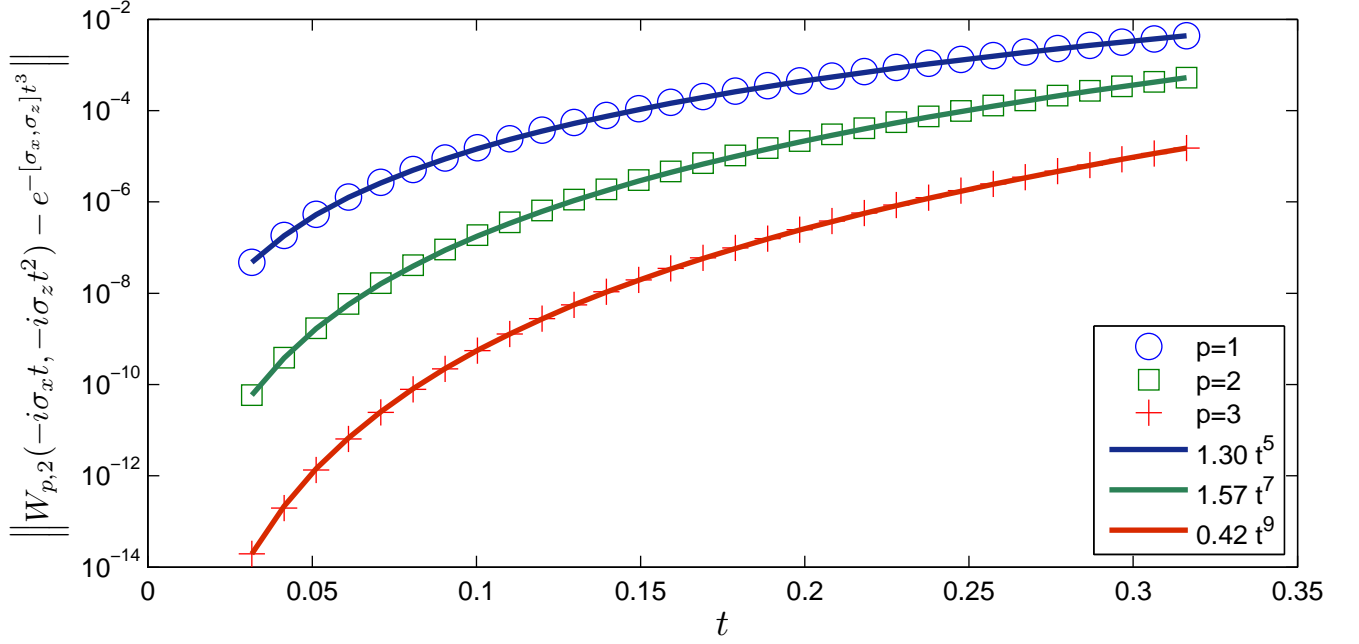
FIG. 2. Error scaling of the formulas $W_{p,2}(-i\sigma_x t, -i\sigma_z t^2)$ for $p = 1, 2, 3$ as approximations of $e^{-[\sigma_x, \sigma_z]t^3}$.

similarly. In both cases, the resulting product formula is denoted $\mathcal{U}_p(A_k t, \ldots, A_0 t)$. In other words, we approximate a single commutator with $\mathcal{U}_p(A_1 t, A_0 t) := V'_{p,1}(A_1 t, A_0 t)$, and we recursively define $\mathcal{U}_p$ for $q > 1$ as

$$\mathcal{U}_p(A_q t, \ldots, A_0 t) := \text{Rep}\left(\left\{ \begin{array}{ll} W_{p,q}(A_q t, Z_{q-1} t^q) & q \text{ even} \\ V_{p,q}(A_q t, Z_{q-1} t^q) & q \text{ odd} \end{array} \right\}, e^{Z_{q-1}(\lambda t)^q} \to \mathcal{U}_p(A_{q-1} \lambda t, \ldots, A_0 \lambda t) \right) \tag{32}$$

for all $q > 1$, where $\text{Rep}(x, a \to b)$ indicates that we replace every instance of a in x with b. For example,

$$\begin{aligned} \mathcal{U}_1(A_2 t, A_1 t, A_0 t) &= \text{Rep}(W_{1,2}(A_2 t, Z_1 t^2), e^{Z_1(\lambda t)^2} \to \mathcal{U}_1(A_1 \lambda t, A_0 \lambda t)) \\ &= \text{Rep}(e^{A_2 \xi_1 t} e^{Z_1(\xi_1 t)^2} e^{2A_2 \xi_1 t} e^{-Z_1(\xi_1 t)^2} e^{A_2 \xi_1 t}, e^{Z_1(\lambda t)^2} \to \mathcal{U}_1(A_1 \lambda t, A_0 \lambda t)) \\ &= e^{A_2 \xi_1 t} \mathcal{U}_1(A_1 \xi_1 t, A_0 \xi_1 t) e^{2A_2 \xi_1 t} \mathcal{U}_1(A_1 \xi_1 t, A_0 \xi_1 t)^{-1} e^{A_2 \xi_1 t}. \end{aligned} \tag{33}$$

At first glance, it may seem surprising that a recursive approximation to $e^{Z_k t^{k+1}}$ can use approximations with error of order less than $2p + k + 1$ (rather than using approximations with error $O(t^{2p+k+1})$ at each stage in the recursion) when the desired overall error is $O(t^{2p+k+1})$. However, since a term with relatively large error is multiplied by a relatively small term, this recursive process straightforwardly yields formulas with error $O(t^{2p+k+1})$.

Lemma 6. *For any positive integer k, let $A_0, \ldots, A_k$ be bounded operators, define Z_k by equation (31), and let $\mathcal{U}_p(A_k t, \ldots, A_0 t)$ be the product formula defined above. Then $\mathcal{U}_p(A_k t, \ldots, A_0 t) = e^{Z_k t^{k+1} + O(t^{2p+k+1})}$.*

Proof. We use induction on k. By Corollary 3, $\mathcal{U}_p(A_1 t, A_0 t) = e^{Z_1 t^2 + O(t^{2p+2})}$, establishing the base case. For the induction step, Theorem 2 and Theorem 5 imply that

$$\begin{aligned} \mathcal{U}_p(A_q t, \ldots, A_0 t) &= \exp([A_q t, Z_{q-1} t^q + O(t^{2p+q})]) \\ &= \exp(Z_q t^{q+1} + O(t^{2p+q+1})). \end{aligned} \tag{34}$$

The desired result then follows by induction. □

To calculate the number of exponentials $N_{p,k}$ appearing in $\mathcal{U}_p(A_k t, \ldots, A_0 t)$, recall from Section II that $V'_{p,1}$ uses $N_{p,1} = 8 \times 6^{p-1}$ exponentials. If $k > 1$ is odd, then $V_{p,k}$ uses $4 \times 6^{p-1}$ exponentials, half of which require further

expansion into $N_{p,k-1}$ exponentials. If k is even, then $W_{p,k}$ uses 5^p exponentials, $2/5$ of which require further expansion into $N_{p,k-1}$ exponentials. Thus, for $k > 1$, we have

$$N_{p,k} = \begin{cases} 5^{p-1}(3 + 2N_{p,k-1}) & k \text{ even} \\ 2 \times 6^{p-1}(1 + N_{p,k-1}) & k \text{ odd.} \end{cases} \tag{35}$$

While it is cumbersome to present the solution in closed form, it is easy to see that for fixed k we have $N_{p,k} = O(6^{pk})$ since we increase the number of exponentials by a factor of $O(6^p)$ with each iteration.

The approach described above uses both odd-k and even-k product formulas. It is possible to derive similar formulas using only odd-k product formulas, but their error bounds are more difficult to prove and the resulting approximations are less efficient.

IV. ERROR BOUNDS FOR NESTED COMMUTATORS

So far, we have presented product formulas that approximate the evolution according to a (nested) commutator to arbitrarily high order. We now provide simple upper bounds on the approximation error that results when using these formulas to simulate evolution for a sufficiently small time. We use the following notation to denote the remainder of a Taylor series expansion truncated at order $\nu - 1$.

Definition 2. *Let $f(t)$ be a bounded operator for any $t \in \mathbb{R}$ with a Taylor series $f(t) = \sum_{n=0}^{\infty} a_n t^n$. Then*

$$\mathbf{R}_\nu(f(t)) := \sum_{n=\nu}^{\infty} a_n t^n. \tag{36}$$

We use the following standard result (with a proof included for completeness).

Lemma 7. *Let $\{A_q \colon q = 1, \ldots, N\}$ be a set of bounded operators and let $\|\cdot\|$ be a submultiplicative norm. Then for any positive integer ν,*

$$\left\| \mathbf{R}_\nu \left(\prod_{q=1}^{N} e^{A_q t} \right) \right\| \leq \mathbf{R}_\nu \left(e^{\sum_{q=1}^{N} \|A_q\| t} \right). \tag{37}$$

Proof. Since each A_q is bounded, Taylor's theorem implies that

$$\left\| \mathbf{R}_\nu \left(\prod_{q=1}^{N} e^{A_q t} \right) \right\| = \left\| \mathbf{R}_\nu \left(\prod_{q=1}^{N} \sum_{p=0}^{\infty} \frac{(A_q t)^p}{p!} \right) \right\|. \tag{38}$$

The submultiplicativity of $\|\cdot\|$ and the triangle inequality imply that

$$\left\| \mathbf{R}_\nu \left(\prod_{q=1}^{N} \sum_{p=0}^{\infty} \frac{(A_q t)^p}{p!} \right) \right\| \leq \mathbf{R}_\nu \left(\prod_{q=1}^{N} \left\| \sum_{p=0}^{\infty} \frac{(A_q t)^p}{p!} \right\| \right)$$

$$\leq \mathbf{R}_\nu \left(\prod_{q=1}^{N} \sum_{p=0}^{\infty} \frac{(\|A_q\| t)^p}{p!} \right)$$

$$= \mathbf{R}_\nu \left(e^{\sum_{q=1}^{N} \|A_q\| t} \right) \tag{39}$$

as claimed. $\qquad \square$

The following lemma establishes general error bounds for exponentials of (possibly nested) commutators.

Lemma 8. *Let $\{A_j \colon j = 0, \ldots, k\}$ be a set of bounded operators and define Z_k by equation (31). Suppose that*

1. *$\mathcal{U}_p(A_k t, \ldots, A_0 t) = e^{Z_j t^{k+1} + O(t^\nu)}$ for some integer $\nu > k + 1$,*
2. *$\mathcal{U}_p(A_k t, \ldots, A_0 t) = \prod_{q=1}^{N_{p,k}} e^{\lambda_q A_{j_q} t}$ where $N_{p,k}^{-1} \sum_{q=1}^{N_{p,k}} |\lambda_q| \leq Q_{p,k}$,*
3. *$\Lambda \geq 2\|A_j\|$ for all $j = 0, \ldots, k$,*

4. $\Lambda t \leq \frac{\ln(2)}{N_{p,k}Q_{p,k}}$, and

5. $N_{p,k}Q_{p,k} \geq 1$.

Then

$$\left\| e^{Z_k t^{k+1}} - \mathcal{U}_p(A_k t, \ldots, A_0 t) \right\| \leq \left(\frac{e N_{p,k} Q_{p,k} \Lambda t}{\nu^{1/(k+1)}} \right)^\nu. \tag{40}$$

Proof. By assumption 1 and the triangle inequality, we have

$$\| e^{Z_k t^{k+1}} - \mathcal{U}_p(A_k t, \ldots, A_0 t) \| \leq \| \mathbf{R}_\nu(e^{Z_k t^{k+1}}) \| + \| \mathbf{R}_\nu(\mathcal{U}_p(A_k t, \ldots, A_0 t)) \|. \tag{41}$$

The proof of the lemma follows from upper bounds for both of these terms.

We first bound $\| \mathbf{R}_\nu(\mathcal{U}_p(A_k t, \ldots, A_0 t)) \|$. Assumptions 2 and 3, together with Lemma 7, imply that

$$\| \mathbf{R}_\nu(e^{\lambda_1 A_1 t} e^{\lambda_2 A_2 t} \ldots) \| \leq \mathbf{R}_\nu \left(e^{\sum_{q=1}^{N_{p,k}} \lambda_q \|A_q\| t} \right) \leq \mathbf{R}_\nu(e^{N_{p,k} Q_{p,k} \Lambda t}), \tag{42}$$

so Taylor's theorem gives

$$\begin{aligned} \| \mathbf{R}_\nu(\mathcal{U}_p(A_k t, \ldots, A_0 t)) \| &\leq \frac{(N_{p,k} Q_{p,k} \Lambda t)^\nu}{\nu!} e^{N_{p,k} Q_{p,k} \Lambda t} \\ &\leq \frac{2(N_{p,k} Q_{p,k} \Lambda t)^\nu}{\nu!}, \end{aligned} \tag{43}$$

where we have used assumption 4 to simplify the bound. A standard variant of Stirling's formula, namely [18]

$$n! \geq \sqrt{2\pi n}\, n^n e^{-n}, \tag{44}$$

gives

$$\| \mathbf{R}_\nu(\mathcal{U}_p(A_k t, \ldots, A_0 t)) \| \leq \frac{2}{\sqrt{2\pi\nu}} \left(\frac{e N_{p,k} Q_{p,k} \Lambda t}{\nu} \right)^\nu. \tag{45}$$

The corresponding bound for the remainder term of $e^{Z_k t^{k+1}}$ follows similarly:

$$\left\| \mathbf{R}_\nu \left(e^{Z_k t^{k+1}} \right) \right\| \leq \frac{(\|Z_k\|^{1/(k+1)} t)^{(k+1)\lceil \frac{\nu}{k+1} \rceil}}{\lceil \frac{\nu}{k+1} \rceil!} e^{\|Z_k\| t^{k+1}}. \tag{46}$$

Since $\|Z_k\| t^{k+1} \leq (\Lambda t)^{k+1} \leq \ln 2$ (by assumptions 3, 4, and 5), we have

$$\begin{aligned} \left\| \mathbf{R}_\nu \left(e^{Z_k t^{k+1}} \right) \right\| &\leq \frac{2(\Lambda t)^{(k+1)\lceil \frac{\nu}{k+1} \rceil}}{\lceil \frac{\nu}{k+1} \rceil!} \\ &\leq \frac{2}{\sqrt{2\pi\nu/(k+1)}} \left(\frac{e^{1/(k+1)} \Lambda t}{(\nu/(k+1))^{1/(k+1)}} \right)^\nu \end{aligned} \tag{47}$$

where the second step uses (44) and $\Lambda t \leq 1$ (by assumptions 4 and 5). To put this in a similar form to (45), we use assumption 5 to find

$$\left\| \mathbf{R}_\nu \left(e^{Z_k t^{k+1}} \right) \right\| \leq \frac{2}{\sqrt{2\pi\nu/(k+1)}} \left(\frac{k+1}{e^k} \right)^{\nu/(k+1)} \left(\frac{e N_{p,k} Q_{p,k} \Lambda t}{\nu^{1/(k+1)}} \right)^\nu. \tag{48}$$

Since $(k+1)/e^k < 1$ for $k \geq 1$ and $\nu/(k+1) > 1$ by assumption 1, we have

$$\sqrt{k+1} \left(\frac{k+1}{e^k} \right)^{\nu/(k+1)} < \frac{(k+1)^{3/2}}{e^k}, \tag{49}$$

so

$$\left\| \mathbf{R}_\nu \left(e^{Z_k t^{k+1}} \right) \right\| \leq \frac{2}{\sqrt{2\pi\nu}} (k+1)^{3/2} e^{-k} \left(\frac{e N_{p,k} Q_{p,k} \Lambda t}{\nu^{1/(k+1)}} \right)^\nu. \tag{50}$$

For $k \geq 1$ we have $(k+1)^{3/2}e^{-k} \leq 2^{3/2}/e < 1.05$, so

$$\|e^{Z_k t^{k+1}} - \mathcal{U}_p(A_k t, \ldots, A_0 t)\| \leq \frac{2}{\sqrt{2\pi\nu}} \left(\frac{eN_{p,k}Q_{p,k}\Lambda t}{\nu}\right)^\nu + \frac{2.1}{\sqrt{2\pi\nu}} \left(\frac{eN_{p,k}Q_{p,k}\Lambda t}{\nu^{1/(k+1)}}\right)^\nu$$

$$\leq \frac{4.1}{\sqrt{2\pi\nu}} \left(\frac{eN_{p,k}Q_{p,k}\Lambda t}{\nu^{1/(k+1)}}\right)^\nu \tag{51}$$

where the second step uses assumption 5. Since $4.1/\sqrt{2\pi\nu} \leq 4.1/\sqrt{6\pi} < 1$, the result follows. $\qquad\square$

The error bound of Lemma 8 can be used to estimate the maximum size of a time step for product formulas given by Theorem 2 for $k = 1$ or any of the product formulas considered in Lemma 6, given upper bounds for $N_{p,k}Q_{p,k}$. Recall from the previous section that for constant k, $N_{p,k} = O(6^{pk})$. Furthermore, since r_p and s_p are monotonically decreasing functions of p (for fixed k), it can easily be seen that they scale as $1/2 + o(1)$ and $1/4 + o(1)$, respectively. This similarly implies that $\beta_p = \gamma_p = (1/2)^{1/(k+1)} + o(1)$ and $\mu_p = \nu_p = (1/3)^{1/(k+1)} + o(1)$ for fixed k. Since $Q_{p,k}$ is the maximum value of products of such terms, it follows that $Q_{p,k} \in O(1)$ and thus $N_{p,k}Q_{p,k} \in O(6^{pk})$. In fact, a more detailed analysis shows that $N_{p,k}Q_{p,k} \in O((2\sqrt{6})^{pk})$, but this does not significantly improve our results.

Note that the bound of Lemma 8 does *not* directly apply to product formulas that result from using $k > 1$ in Theorem 2, or any of the formulas considered in Theorem 5, because such product formulas contain varying powers of t. Error bounds for such cases could be proved by similar reasoning, but are unlikely to give bounds that are as tight, and the cases covered by Lemma 8 are more relevant for our applications.

V. NEAR-LINEAR SCALING WITH TOTAL EVOLUTION TIME

Lemma 8 bounds the distance between the true evolution $U(t)$ and the approximation $\mathcal{U}_p(A_k t, \ldots, A_0 t)$. Although the formulas are very accurate for small t, the error (as measured by a sub-multiplicative norm such as the spectral norm) can be large if $\Lambda t > 1$. To construct a product formula approximation that is close to $U(t)$ for general Λt, we divide the evolution into short segments and show that the cumulative error from these segments is at most ϵ.

To ensure additivity of approximation errors from each time step, we restrict ourselves to the case where the operators A_j are anti-Hermitian (as in quantum mechanics), a restriction that was unnecessary in previous sections. By carefully choosing p to be a function of t and ϵ, the number of exponentials needed to approximate $U(t)$ within error ϵ scales only slightly superlinearly with t^{k+1} (the total evolution time) and sub-polynomially with $1/\epsilon$.

The following lemma is the main result of this section. It provides an upper bound for the minimum number of time steps, r, needed to ensure that the cumulative approximation errors from all r time steps sum to at most ϵ.

Lemma 9. *Let $\{A_j : j = 1, \ldots, k\}$ be a set of bounded anti-Hermitian operators and let $\epsilon > 0$. If Assumptions 3, 1, and 2 of Lemma 8 are satisfied for $\mathcal{U}_p(A_k t/r^{1/(k+1)}, \ldots, A_0 t/r^{1/(k+1)})$, where the integer r satisfies*

$$r \geq \frac{\left(\frac{eN_{p,k}Q_{p,k}\Lambda t}{(2p+k+1)^{1/(k+1)}}\right)^{k+1+\frac{(k+1)^2}{2p}}}{\epsilon^{(k+1)/(2p)}} \tag{52}$$

and the error ϵ satisfies

$$\epsilon \leq \left(\frac{e}{(2p+k+1)^{1/(k+1)}}\right)^{2p+k+1} \ln(2)^{2p} \left(N_{p,k}Q_{p,k}\Lambda t\right)^{k+1}, \tag{53}$$

then $\|U(t) - \mathcal{U}_p(A_k t/r^{1/(k+1)}, \ldots, A_0 t/r^{1/(k+1)})^r\| \leq \epsilon$.

Proof. Since the operators A_j are anti-Hermitian, $e^{A_j t}$ and $e^{Z_k t^{k+1}}$ are unitary. Errors in unitary operations are subadditive, so the total error is at most the number of time steps used in the simulation, r, times the error in simulating the commutator exponential during each time step. With r time steps, the duration of each step is $t/r^{1/(k+1)}$. By Lemma 8, the error in approximating $e^{Z_k t^{k+1}}$ is at most ϵ if

$$\left(\frac{eN_{p,k}Q_{p,k}\Lambda t}{(2p+k+1)^{1/(k+1)}r^{1/(k+1)}}\right)^{2p+k+1} \leq \frac{\epsilon}{r}, \tag{54}$$

i.e., if

$$\frac{1}{\epsilon}\left(\frac{eN_{p,k}Q_{p,k}\Lambda t}{(2p+k+1)^{1/(k+1)}}\right)^{2p+k+1} \leq r^{(2p)/(k+1)}, \tag{55}$$

which is equivalent to the stated condition (52) on r.

The argument that the restriction on ϵ in (53) implies assumptions 5 and 4 of Lemma 8 is straightforward. We first demonstrate that $N_{p,k}Q_{p,k} \geq 1$ for $\mathcal{U}_p(A_k t/r^{1/(k+1)}, \ldots, A_1 t/r^{1/(k+1)})$ and then use this fact to show the validity of assumption 4. We show this by proving loose lower bounds for $N_{p,k}$ and $Q_{p,k}$ and showing that the product of these lower bounds yields a result greater than 1. First, we use (35) to see that

$$N_{p,k} \geq 2 \cdot 5^{p-1} N_{p,k-1} \geq (2 \cdot 5^{p-1})^k, \tag{56}$$

where the last inequality follows from solving the prior recursion relation using $N_{p,1} \geq 2 \cdot 5^{p-1}$ as the initial condition.

Since $Q_{p,k} \geq \max_j\{|\lambda_j|\}$, where λ_j is the duration of the jth exponential in $\mathcal{U}_p(A_k t/r^{1/(k+1)}, \ldots, A_1 t/r^{1/(k+1)})$ divided by $t/r^{1/(k+1)}$, a lower bound on $|\lambda_j|$ is also a lower bound for $Q_{p,k}$. The form of $\mathcal{U}_p(A_k t/r^{1/(k+1)}, \ldots, A_1 t/r^{1/(k+1)})$ requires that we recursively use the results of Theorem 2 and Theorem 5. Note that this recursion must be used k times and each such formula is constructed by p applications of the recursive relations in Theorem 2 or Theorem 5. Theorem 2 and Theorem 5 imply that each λ_j must be a product of at most qk terms of the form ξ_k, μ_ℓ, ν_ℓ, β_ℓ, and γ_ℓ for $\ell \in \{1, \ldots, p\}$. Because μ_ℓ, ν_ℓ, β_ℓ, and γ_ℓ are monotonically decreasing functions of ℓ, taking their limit as $\ell \to \infty$ shows that $\max\{\mu_\ell, \nu_\ell, \beta_\ell, \gamma_\ell\} \geq 1/3^{1/(k+1)} \geq 1/\sqrt{3}$. Furthermore, $\xi_k = 2^{-1/(k+1)} \geq 1/\sqrt{3}$. Because λ_j is a product of qk such terms, it follows that

$$Q_{p,k} \geq \min_j |\lambda_j| \geq 3^{-kp/2}. \tag{57}$$

Combining (56) and (57) gives

$$N_{p,k}Q_{p,k} \geq \frac{(2 \cdot 5^{p-1})^k}{3^{kp/2}} \geq \frac{2}{\sqrt{3}} > 1. \tag{58}$$

Since $N_{p,k}Q_{p,k} \geq 1$, assumption 4 of Lemma 8 is implied by

$$\frac{\Lambda t}{r^{1/(k+1)}} \leq \frac{\ln(2)}{N_{p,k}Q_{p,k}}. \tag{59}$$

Using our lower bound on r in (52), we find that (59) is implied by

$$\frac{\Lambda t \epsilon^{1/(2p)}}{\left(\frac{e N_{p,k}Q_{p,k}\Lambda t}{(2p+k+1)^{1/(k+1)}}\right)^{1+\frac{k+1}{2p}}} \leq \frac{\ln(2)}{N_{p,k}Q_{p,k}}, \tag{60}$$

which is equivalent to (53) after solving for ϵ and simplifying the resulting expression. $\square$

Rather than fixing p and choosing r such that the error is at most ϵ, a more cost-effective strategy is to choose p to minimize the total number of exponentials. The required number of exponentials to achieve error at most ϵ is

$$N_{\text{exp}} = N_{p,k}r \geq N_{p,k}\frac{\left(\frac{e N_{p,k}Q_{p,k}\Lambda t}{(2p+k+1)^{1/(k+1)}}\right)^{k+1+\frac{(k+1)^2}{2p}}}{\epsilon^{(k+1)/(2p)}}. \tag{61}$$

This expression is minimized for some p due to two competing tendencies: $\epsilon^{-(k+1)/(2p)}$ shrinks as p increases, whereas $N_{p,k}$ grows exponentially in p for k constant, as discussed at the end of Section III. We can approximate the optimal value of p by equating two terms:

$$N_{p,k}^{k+2} = \left(\frac{\Lambda t}{\epsilon^{1/(k+1)}}\right)^{(k+1)^2/(2p)}. \tag{62}$$

Here we neglect the term $N_{p,k}^{(k+1)^2/(2p)}$ since it is approximately constant for $N_{p,k} \in e^{\Theta(p)}$. We could also include the detailed behavior of $Q_{p,k}$ to more accurately estimate the optimal value of p, but doing so complicates the discussion and does not qualitatively change the result.

Taking logarithms of both sides of (62), using $N_{p,k} \in e^{\Theta(p_{\text{opt}})}$, and considering constant k, we find

$$p_{\text{opt}} \in \Theta\left(\sqrt{\log\left(\frac{\Lambda t}{\epsilon}\right)}\right). \tag{63}$$

Since p_{opt} is sublogarithmic in $\Lambda t/\epsilon$, we find that $N_{p,k}Q_{p,k}$ grows subpolynomially. Thus, by choosing $p = p_{\text{opt}}$, the number of exponentials used to approximate $e^{Z_k t^{k+1}}$ scales as

$$N_{\exp} \in (\Lambda t)^{k+1} \left(\frac{\Lambda t}{\epsilon}\right)^{o(1)}. \tag{64}$$

Similarly to the simulation of sums of Hamiltonians using Suzuki formulas [5, 6], this shows that the cost of simulating an operator exponential scales only slightly superlinearly with the total evolution time t^{k+1} and subpolynomially with $1/\epsilon$. (Note that the analogous scaling for the Suzuki formulas [16] follows upon substituting $k = 0$.) In the next section we show that this behavior is nearly optimal.

VI. OPTIMALITY PROOF VIA QUANTUM SEARCH

We have shown that high-order approximations to exponentials of commutators can be constructed with a number of exponentials that scales almost linearly with the evolution time. Here we show that such schemes are nearly optimal by demonstrating that a simulation using a sublinear number of exponentials would violate the lower bound for quantum search [19]. The argument is similar to the analysis of the continuous-time algorithm for quantum search [20], but uses a Hamiltonian expressed as a commutator instead of as a sum.

Theorem 10. *Any product formula approximation for $e^{[A,B]T}$ for generic A, B consists of $\Omega(T)$ exponentials.*

Proof. Consider the problem of searching for an unknown $w \in \{1, \ldots, n\}$ using a black box acting as

$$|x, b\rangle \mapsto \begin{cases} |x, b\rangle & \text{if } x \neq w \\ |x, \bar{b}\rangle & \text{if } x = w. \end{cases} \tag{65}$$

A quantum computer must make $\Omega(\sqrt{n})$ queries to the black box to solve this problem with bounded error [19].

To solve the search problem, we can begin in the state $|+\rangle := \frac{1}{\sqrt{n}} \sum_{x=1}^{n} |x\rangle$ and try to evolve with the Hamiltonian $i[|w\rangle\langle w|, |+\rangle\langle +|]$ for some time T such that

$$e^{[|w\rangle\langle w|, |+\rangle\langle +|]T} |+\rangle \approx |w\rangle \tag{66}$$

for any $w \in \{1, \ldots, n\}$.

The Hamiltonian $i[|w\rangle\langle w|, |+\rangle\langle +|]$ generates a rotation in the plane spanned by $|w\rangle$ and $|+\rangle$. We therefore choose our basis to contain $|w\rangle$ and $|\perp\rangle := \frac{1}{\sqrt{n-1}} \sum_{x \neq w} |x\rangle$. Let

$$\begin{aligned} Y &:= \sqrt{n}[|w\rangle\langle w|, |+\rangle\langle +|] \\ &= |w\rangle\langle +| - |+\rangle\langle w|. \end{aligned} \tag{67}$$

The action of Y on the basis states is

$$Y |w\rangle = -\sqrt{\frac{n-1}{n}} |\perp\rangle \tag{68}$$

$$Y |\perp\rangle = \sqrt{\frac{n-1}{n}} |w\rangle. \tag{69}$$

This is simply $i\sqrt{(n-1)/n}$ times the Pauli operator σ_y on the relevant subspace, with eigenvectors

$$|e_\pm\rangle = \frac{|w\rangle \pm i |\perp\rangle}{\sqrt{2}} \tag{70}$$

and corresponding eigenvalues

$$e_\pm = \pm i\sqrt{\frac{n-1}{n}}. \tag{71}$$

We have

$$|w\rangle = \frac{|e_+\rangle + |e_-\rangle}{\sqrt{2}} \tag{72}$$

$$|+\rangle = \frac{|e_+\rangle}{\sqrt{2}}\left(\frac{1}{\sqrt{n}} - i\sqrt{\frac{n-1}{n}}\right) + \frac{|e_-\rangle}{\sqrt{2}}\left(\frac{1}{\sqrt{n}} + i\sqrt{\frac{n-1}{n}}\right). \tag{73}$$

Defining ϕ so that $\tan\phi = \sqrt{n-1}$, the time-evolved state is

$$e^{\frac{Yt}{\sqrt{n}}}|+\rangle = \frac{|e_+\rangle}{\sqrt{2}}e^{i\left(\sqrt{\frac{n-1}{n}}\frac{T}{\sqrt{n}} - \phi\right)} + \frac{|e_-\rangle}{\sqrt{2}}e^{-i\left(\sqrt{\frac{n-1}{n}}\frac{T}{\sqrt{n}} - \phi\right)}. \tag{74}$$

The evolution then yields the desired state, $|w\rangle$, when the phases of both eigenvectors are the same. This first occurs when

$$T = \frac{\phi n}{\sqrt{n-1}} = \frac{\pi}{2}\sqrt{n} + O(1). \tag{75}$$

The exponential of $|w\rangle\langle w|$ for any time can be simulated with 2 queries to the black box, and the exponential of $|+\rangle\langle+|$ for any time can be simulated without querying the black box. Thus a product formula approximation of $e^{[|w\rangle\langle w|, |+\rangle\langle+|]T}$ using $N_{\exp}$ exponentials gives an algorithm for the search problem using $O(N_{\exp})$ queries. By the $\Omega(\sqrt{n})$ lower bound on search, we find that the number of exponentials in the product formula must be $N_{\exp} \in \Omega(\sqrt{n}) = \Omega(T)$. □

Thus there is no product formula approximation to $e^{[A,B]t^{k+1}}$, for generic A and B, that consists of $o((\Lambda t)^{k+1})$ exponentials. Equation (64) shows that our high-order product formula approximations use $(\Lambda t)^{k+1+o(1)}$ exponentials, so they are nearly optimal.

VII. APPLICATIONS

A. Controlling Quantum Systems

Quantum control is the art of designing sequences of quantum operations (often referred to as pulses) that approximate a desired operation. This is especially useful when the target operation cannot be directly implemented. Techniques that are used to design pulse sequences include [14] Lie–Trotter–Suzuki sequences, Dyson series, Fer and Wilcox expansions, Solovay–Kitaev sequences, and the Magnus series. The latter three techniques explicitly involve implementing exponentials of commutators, which can be turned into pulse sequences using product formulas. Current approaches to implement such commutator exponentials are typically inaccurate and often must be numerically optimized via a method such as GRAPE [21, 22], which uses gradient ascent to numerically optimize the efficiency and robustness of a pulse sequence that implements a desired unitary operation. The quality of the numerically optimized sequence depends on the initial input sequence, so our results may be useful for designing pulse sequences (and perhaps understanding why a numerically optimized control sequence takes a particular form).

A simple application of our results to quantum control addresses the problem of performing an arbitrary single-qubit rotation given access to two non-orthogonal rotation axes. Consider the single-qubit Hamiltonian

$$H(t) = B_0\sigma_z + B_c(t)(\sigma_z + \sigma_x) \tag{76}$$

where $B_c(t)$ is a controllable magnetic field that can switch between the values 0 and $B_c < B_0$. The goal is to perform the operation

$$U(t) = e^{-i\omega_0\sigma_y t^2} \tag{77}$$

for some constant ω_0, which is sufficient to establish universal control of the qubit if used in concert with the z rotation provided by the Hamiltonian when $B_c = 0$. We can achieve the desired y rotation as

$$U(t) = e^{-\omega_0[\sigma_z,\sigma_x]t^2/2} = e^{-\omega_0[B_0\sigma_z, B_0\sigma_z + B_c(\sigma_z+\sigma_x)]t^2/(2B_0B_c)}. \tag{78}$$

For simplicity, define

$$A := B_0\sigma_z,$$
$$B := B_0\sigma_z + \frac{\omega_0}{2B_0}(\sigma_z + \sigma_x), \tag{79}$$

which can both be realized by appropriate choices of B_c. Then $[A, B] = i\omega_0\sigma_y$. Corollary 3 shows that we can approximate $U(t)$ as

$$U(t) = e^{-iAt/\sqrt{2}}e^{-iBt/\sqrt{2}}e^{iAt/\sqrt{2}}e^{iBt/\sqrt{2}}e^{iAt/\sqrt{2}}e^{iBt/\sqrt{2}}e^{-iAt/\sqrt{2}}e^{-iBt/\sqrt{2}} + O(t^4). \tag{80}$$

Theorem 2 can be used to construct higher-order variants of this expression.

Note that we have assumed that we can implement evolutions of the form $\exp(iAt)$ and $\exp(iBt)$ for $t > 0$, corresponding to evolution for negative time. For this single-qubit system, this can easily be done by evolving under A or B for some positive time. However, this may not be as straightforward in general.

Recent work by Borneman, Granade, and Cory [23] further highlights the role of commutator exponentials in quantum control by using them to suppress undesirable interactions in coupled spin systems and introduce new couplings that are not present in the original Hamiltonian. Their work constructs a nested commutator of the form $e^{i[A,[A,B]]t^3}$ using only the ability to implement exponentials of A and B individually. They achieve this by presenting a product formula for nested commutators that is outside the scope of our formalism (since it exploits the fact that two of the terms in the nested commutator are equal), namely

$$W_{1/2,2}^{\mathrm{BGC}}(-iAt, -iBt) := e^{-iAt}e^{-iBt}e^{iAt}e^{iBt}e^{iAt}e^{-iBt}e^{-iAt}e^{iBt}. \tag{81}$$

They show that this product formula obeys

$$\|W_{1/2,2}^{\mathrm{BGC}}(-iAt, -iBt) - e^{i[A,[A,B]]t^3}\| \in O(t^4). \tag{82}$$

Here we use the subscript $p = 1/2$ since, with $k = 2$, the error is $O(t^4) = O(t^{2p+k+1})$ as in our previous product formulas. The validity of (82) follows directly from

$$\ln(e^{-iAt}e^{-iBt}e^{iAt}e^{iBt}) = -[A,B]t^2 + \frac{it^3}{2!}[A+B,[A,B]] + \frac{t^4}{3!}([A,[B,[B,A]]]/2 + [A+B,[A+B,[A,B]]]) + O(t^5) \tag{83}$$

and the Baker–Campbell–Hausdorff formula because all terms of $O(t^3)$ cancel in (81) except for $it^3[A,[A,B]]$.

Note that $W_{1/2,2}^{\mathrm{BGC}}(-iAt, -iBt)$ is actually correct to $O(t^5)$ if $[A,[B,[B,A]]] = 0$, which holds, for example, when A and B are Pauli operators. We refer to the BGC formula as $W_{1,2}^{\mathrm{BGC}}(-iAt, -iBt)$ in such cases to reflect the fact that the error is $O(t^5)$. Also note that by a similar calculation to (83),

$$W_{1/2,2}^{\mathrm{BGC}}(-iAt, -iBt)^{-1} = e^{-iBt}e^{iAt}e^{iBt}e^{-iAt}e^{-iBt}e^{-iAt}e^{iBt}e^{iAt} \tag{84}$$

satisfies

$$\|W_{1/2,2}^{\mathrm{BGC}}(-iAt, -iBt)^{-1} - e^{-i[A,[A,B]]t^3}\| \in O(t^4). \tag{85}$$

This error term is also $O(t^5)$ when $[A,[B,[A,B]]] = 0$.

Even though the BGC formula goes beyond our formalism, we can still use our results to refine it into higher-order versions. It is not immediately clear how to do this using Theorem 5 since the proof of that theorem relies on determining the parity of the number of As in a particular error term. For the error term of order t^5 in the BGC formula, we cannot determine the parity of the number of As because both A and B are associated with t rather than with t and t^2, respectively.

Nevertheless, it turns out that the construction described in Theorem 5 refines $W_{1/2,2}^{\mathrm{BGC}}(-iAt, -iBt)$ into a formula denoted $W_{1,2}^{\mathrm{BGC}}(-iAt, -iBt)$ that has error $O(t^5)$ (rather than $O(t^6)$ for the reasons described above). We can then continue to refine the formula to arbitrarily high order by applying the construction of Theorem 5 recursively. In particular, we have the following.

Corollary 11. *Let $W_{p,2}^{\mathrm{BGC}}(At, Bt)$ be a product formula approximation to $\exp([A,[A,B]]t^3)$ that has error $O(t^{2p+3})$. Define*

$$W_{p+1/2,2}^{\mathrm{BGC}}(At, Bt) := W_{p,2}^{\mathrm{BGC}}(A\nu_p t, B\nu_p t)^2 W_{p,2}^{\mathrm{BGC}}(A\mu_p t, B\mu_p t)^{-1}W_{p,2}^{\mathrm{BGC}}(A\nu_p t, B\nu_p t)^2 \tag{86}$$
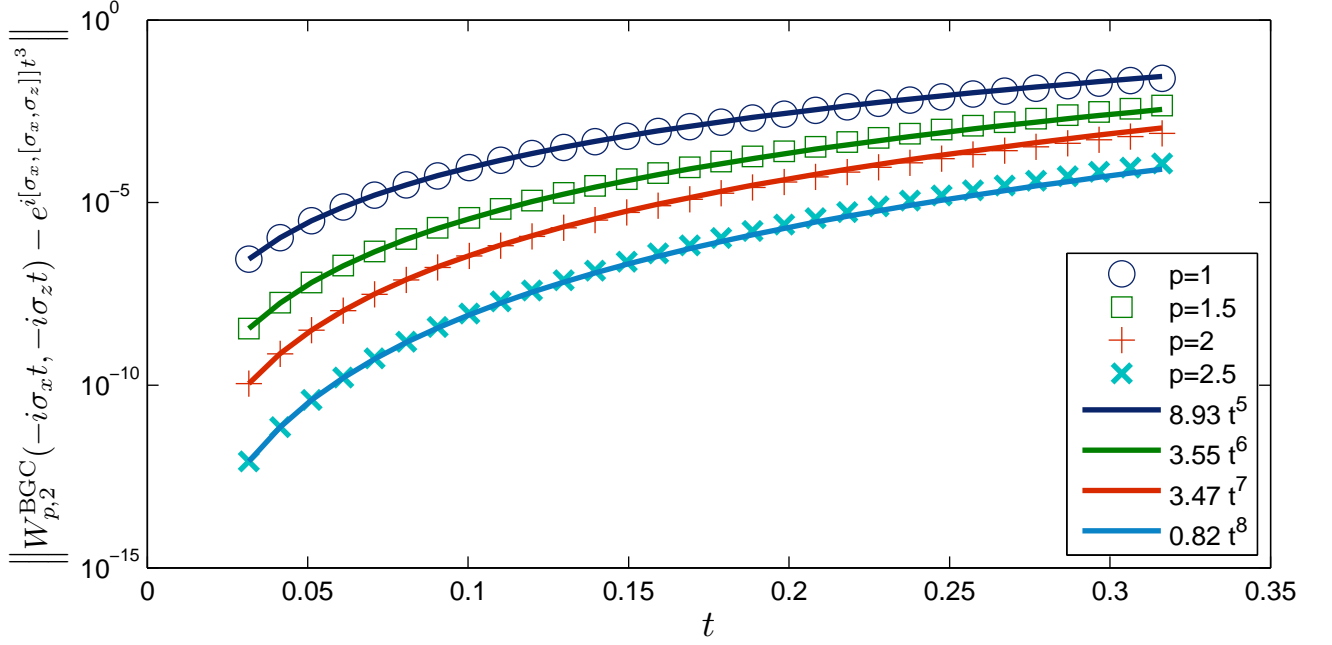
FIG. 3. Error scaling of the formulas $W_{p,2}^{\mathrm{BGC}}(-i\sigma_x t, -i\sigma_z t)$ for $p = 1, 1.5, 2, 2.5$ as approximations of $e^{i[\sigma_x,[\sigma_x,\sigma_z]]t^2}$, where $\|\cdot\|$ is the 2-norm.

where

$$\mu_p := (4s_p)^{1/3}, \qquad \nu_p := (1/4 + s_p)^{1/3}, \qquad s_p := \frac{4^{\frac{3}{2p+3}}}{4\left(4 - 4^{\frac{3}{2p+3}}\right)}. \tag{87}$$

Then $\|W_{p+1/2,2}^{\mathrm{BGC}}(At, Bt) - e^{[A,[A,B]]t^3}\| \in O(t^{2p+4})$.

Proof. We have $W_{p+1/2,2}^{\mathrm{BGC}}(At, Bt) = \exp([A, [A, B]]t^3) + C(A, B)t^{2p+3}$ for some operator $C(A, B)$. Similarly, from (85), $W_{p+1/2,2}^{\mathrm{BGC}}(At, Bt)^{-1} = \exp(-[A, [A, B]]t^3) + \tilde{C}(A, B)t^{2p+3}$ for some operator $\tilde{C}(A, B)$. Since

$$
\begin{aligned}
\mathbb{1} &= W_{p+1/2,2}^{\mathrm{BGC}}(At, Bt)W_{p+1/2,2}^{\mathrm{BGC}}(At, Bt)^{-1} \\
&= \mathbb{1} + (C(A, B) + \tilde{C}(A, B))t^{2p+3} + O(t^{2p+4}),
\end{aligned} \tag{88}
$$

$\tilde{C}(A, B) = -C(A, B)$. Then the result follows from the same calculation as in (27). $\qquad\square$

An example of this generalized BGC formula with $A = i\sigma_x$ and $B = i\sigma_z$ is presented in Figure 3. Here the lowest-order formula is $W_{1,2}^{\mathrm{BGC}}(-i\sigma_x t, -i\sigma_z t)$ since $[A, [B, [B, A]]] = 0$. We see that the upper bound of Corollary 11 is essentially optimal: the approximation error scales as $O(t^{2p+3})$. Note that our higher-order BGC formulas use $O(5^{2p})$ exponentials, an improvement over the $O(6^{2p})$ exponentials appearing in the results of Lemma 6.

As another simple application, we could also use the nested commutator formulas provided by Lemma 6 or Corollary 11 to implement $e^{[A,[A,B]]t^3}$ in the simple example discussed at the beginning of this section, where A and B are given by (79). This produces an effective x rotation, effectively canceling the σ_z terms present in B.

B. Product Formula Approximations for Anticommutators

Another application of our results is to implement Hamiltonian evolution according to anticommutators. The anticommutator of two operators A and B is

$$\{A, B\} := AB + BA. \tag{89}$$

We can implement $e^{\{A,B\}t^2}$ using a commutator simulation by enlarging the Hilbert space. This method can also be used to implement powers of a Hamiltonian, or more generally, products of powers of commuting Hamiltonians, which may be applied to simulate many-body couplings. In contrast to the techniques described in the previous section, this method can be used to introduce interactions that do not arise naturally as commutators of terms already appearing in the Hamiltonian. Our product formula approach for simulating anticommutators can therefore be seen as a complementary method to the generalization of the BGC approach described in Section VII A.

We extend the Hilbert space from $\mathcal{H}$ to $\mathcal{H} \otimes \mathbb{C}^2$ and create analogs of our operators A and B on this enlarged space, namely

$$A' := A \otimes \sigma_y$$
$$B' := B \otimes \sigma_x, \tag{90}$$

where σ_x and σ_y are Pauli operators. Then

$$[A', B'] = AB \otimes \sigma_y \sigma_x - BA \otimes \sigma_x \sigma_y$$
$$= -i\{A, B\} \otimes \sigma_z. \tag{91}$$

Let $|\psi\rangle \in \mathcal{H}$ be any quantum state. Then

$$[A', B'] |\psi\rangle \otimes |0\rangle = -i\{A, B\} |\psi\rangle \otimes |0\rangle, \tag{92}$$

so the action of the anticomutator can be simulated using an ancilla qubit in the eigenstate $|0\rangle$ of σ_z. In particular,

$$e^{-i\{A,B\}t^2} |\psi\rangle \otimes |0\rangle = e^{[A',B']t^2} |\psi\rangle \otimes |0\rangle$$
$$= V'_{p,1}(A't, B't) |\psi\rangle \otimes |0\rangle + O(t^{2p+2}). \tag{93}$$

This shows how to simulate anticommutators using a simulation of commutators. It is straightforward to generalize this construction to simulate nested anticommutators in terms of nested commutators.

While we have already discussed applications of implementing exponentials of commutators, exponentials of anticommutators may seem less natural. However, one application of anticommutators (between commuting operators) is to simulate many-body couplings. For example, consider the one-body operators $A_j = \sigma_z^{(j)}$ for $j = 0, \ldots, k$, where the superscript indicates which spin is acted on. Then

$$\{A_k, \ldots \{A_2, \{A_1, A_0\}\} \ldots\} = 2^k \sigma_z^{\otimes k+1}, \tag{94}$$

so the Hamiltonian $\sigma_z^{\otimes k+1}$ can be simulated using a product of exponentials that each involve only two-body terms, which are typically more natural than k-body terms with $k > 2$. This approach provides an alternative to direct simulation methods for many-body terms [24–27]. Since no explicit changes of basis are performed, this simulation may more accurately reproduce features of the ideal evolution (e.g., its behavior in the presence of errors).

As an example of a complete simulation, consider the toric code Hamiltonian [28]

$$H = -J \left(\sum_v A_v + \sum_p B_p \right), \tag{95}$$

where $A_v := \prod_{i \in v} \sigma_x^{(i)}$ and $B_p := \prod_{j \in p} \sigma_z^{(j)}$ are four-body coupling terms acting on qubits on the edges of a square lattice, where v denotes the set of qubits surrounding a vertex of the lattice and p denotes the set of qubits on the edges of a plaquette. The vertex and plaquette operators commute, so the time-evolution operator is

$$e^{-iHt} = \prod_v e^{iJA_v t} \prod_p e^{iJB_p t}. \tag{96}$$

Each of these exponentials involves a four-body Hamiltonian and therefore can be implemented using only two-body interactions via the approach outlined in (94) with $k = 3$.

The following theorem generalizes the above examples to simulate a Hamiltonian that is a product of powers of commuting matrices and provides a loose upper bound for the scaling of the number of exponentials (which is proportional to the number of quantum operations) needed to simulate the Hamiltonian.

Theorem 12. *Let $\{A_\ell \colon \ell = 1, \ldots, m\}$ be commuting Hermitian matrices, let $\{\alpha_\ell \colon \ell = 1, \ldots, m\}$ be integers, let $k := (\sum_{\ell=1}^{m} \alpha_\ell) - 1$, and let p be a positive integer. Then $\exp(-i2^k A_1^{\alpha_1} \ldots A_m^{\alpha_m} t^{k+1})$ can be simulated with error at most $\epsilon > 0$ using*

$$N_{\exp} \in O\left(\frac{6^{pk} \left(6^{pk} \Lambda t \right)^{k+1+(k+1)^2/(2p)}}{\epsilon^{(k+1)/(2p)}} \right) \tag{97}$$

exponentials, where $\Lambda \geq 2 \max_j \|A_j\|$.

Proof. The proof is a simple generalization of the previous discussion. First introduce operators $\mathcal{A}_j$ for $j = 0, \ldots, k$ such that each $\mathcal{A}_j$ corresponds to one of the operators A_ℓ, where $\mathcal{A}_j = A_\ell$ for α_ℓ values of j. Then define $\mathcal{A}_j'$ as the isometric extension

$$\mathcal{A}_j' := \begin{cases} \mathcal{A}_0 \otimes \sigma_x & \text{if } j = 0 \\ \mathcal{A}_j \otimes \sigma_y & \text{if } j > 0. \end{cases} \tag{98}$$

We now show by induction that the nested commutator of all the $\mathcal{A}_j'$ is an isometric extension of $\mathcal{A}_0 \mathcal{A}_1 \ldots \mathcal{A}_k$. Specifically, we claim that

$$[\mathcal{A}_k', [\ldots, [\mathcal{A}_1', \mathcal{A}_0'] \ldots]] = \begin{cases} -i2^k \prod_{q=1}^{k} \mathcal{A}_q \otimes \sigma_x & k \text{ odd} \\ 2^k \prod_{q=1}^{k} \mathcal{A}_q \otimes \sigma_z & k \text{ even} \end{cases}$$
$$= \begin{cases} -i2^k A_0^{\alpha_0} A_1^{\alpha_1} \cdots A_k^{\alpha_k} \otimes \sigma_x & k \text{ odd} \\ 2^k A_0^{\alpha_0} A_1^{\alpha_1} \cdots A_k^{\alpha_k} \otimes \sigma_z & k \text{ even.} \end{cases} \tag{99}$$

We have already demonstrated the base case in (92) by showing that $[\mathcal{A}_1', \mathcal{A}_0'] = -2i\mathcal{A}_1 \mathcal{A}_0 \otimes \sigma_z$. Thus, assume that the claim holds for some given value of k. Then we have

$$[\mathcal{A}_{k+1}', [\mathcal{A}_k', [\ldots, [\mathcal{A}_1', \mathcal{A}_0'] \ldots]]] = \begin{cases} -i2^{k+1} \prod_{q=1}^{j+1} \mathcal{A}_q \otimes [\sigma_y, \sigma_x] & k \text{ odd} \\ 2^{k+1} \prod_{q=1}^{j+1} \mathcal{A}_q \otimes [\sigma_y, \sigma_z] & k \text{ even} \end{cases}$$
$$= \begin{cases} 2^{k+1} \prod_{q=1}^{j+1} \mathcal{A}_q \otimes \sigma_z & k+1 \text{ even} \\ -i2^{k+1} \prod_{q=1}^{j+1} \mathcal{A}_q \otimes \sigma_x & k+1 \text{ odd,} \end{cases} \tag{100}$$

and the claim follows by induction.

Finally, we simulate the nested commutator exponential by $\mathcal{U}_p(\mathcal{A}_k' t, \ldots, \mathcal{A}_0' t)$ (with an eigenstate of σ_z or σ_x, as appropriate, in the ancilla register). The theorem follows by using (61) for $N_{p,k} \in O(6^{pk})$ and $Q_{p,k} < 1$ (as justified in Section V). □

In the example of the toric code, Theorem 12 implies that e^{-iHt} can be simulated with error at most ϵ using

$$O\left(\frac{6^{3p} \left(6^{3p} (Jt)^{1/4} \right)^{4+8/p} n}{(\epsilon/n)^{2/p}} \right) \tag{101}$$

operations, where n is the number of vertices (or equivalently, the number of plaquettes) in the lattice. We replace ϵ with ϵ/n because simulation errors are subadditive and there are $O(n)$ terms to be simulated in (96). Equation (101) shows that the toric code dynamics can be implemented using a number of two-body interactions that scales near-linearly with t while using only one ancilla qubit (although it may be more convenient to use $O(n)$ ancilla qubits so that nearest-neighbor interactions suffice).

An alternative to Theorem 12 is to estimate the eigenvalues of each A_i using phase estimation and introduce a conditional phase that depends on the eigenvalue (e.g., as described in [29]). The method based on commutators has the advantage that it needs only one ancilla qubit, rather than a logarithmic-size register used to store the estimated phase. Also, phase estimation algorithms have the drawback of requiring $O(1/\epsilon)$ operations in general. In contrast, our approach uses $\epsilon^{-o(1)}$ operations, so Theorem 12 may be useful for high-precision simulations.

VIII.   CONCLUSION

We have presented recursive constructions that approximate exponentials of commutators as products of exponentials of the elementary terms. We provided explicit upper bounds on the approximation error, found upper bounds on the number of elementary exponentials needed to approximate an exponential of commutators to within a fixed error tolerance, and established near-optimality of our results by relating commutator simulation to unstructured search.

Our formulas have natural applications in quantum control, where they can be used to introduce or suppress interactions. Our formulas can be much more accurate than those used in current applications, and consequently may have applications for designing highly accurate control sequences or dynamical decoupling sequences. Such sequences may be valuable as initial guesses in numerical pulse finding methods.

Anticommutators can also be simulated using commutator simulation together with a dilation of the Hilbert space to switch the signs of the terms in the anticommutator. This technique can be used to implement many-body Hamiltonians or simulate powers of Hamiltonians.

This work raises several natural questions. Although our product formula approximations have nearly optimal scaling with t, alternative methods could come closer to or even saturate the lower bound. However, this problem may be difficult as it has long been open in the case of exponentials of sums. A more tractable goal might be to seek improved performance as a function of k, which would be useful since the number of exponentials needed for a high-order approximation for large k can be prohibitively expensive.

Applications of our results to quantum control are reminiscent of the Solovay–Kitaev theorem [30], which is also used to generate pulse sequences. That approach is based on the group commutator, which we also use in Lemma 1. Our expression in (80) has the advantage of more precisely implementing the desired rotation than the group commutators used in Solovay–Kitaev decompositions [31–33]. This suggests that our product formulas may also find application in Solovay–Kitaev algorithms.

Finally, recent work in numerical analysis [34, 35] and quantum simulation [36] has shown that *multi-product formulas*, which are linear combinations of product formulas, can provide more efficient approximations to operator exponentials. It might be interesting to investigate whether similar techniques could lead to more efficient (but possibly less numerically stable) approximations to exponentials of commutators.

ACKNOWLEDGMENTS

[1] A. J. Chorin, T. J. R. Hughes, M. F. McCracken, and J. E. Marsden, *Product formulas and numerical algorithms*, Communications on Pure and Applied Mathematics **31**, 205 (1978).

[2] S. Lloyd, *Universal quantum simulators*, Science **273**, 1073 (1996).

[3] D. Aharonov and A. Ta-Shma, *Adiabatic quantum state generation and statistical zero knowledge*, in *Proceedings of the 35th ACM Symposium on Theory of Computing* (2003), pp. 20–29, arXiv:quant-ph/0301023.

[4] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, *Exponential algorithmic speedup by quantum walk*, in *Proceedings of the 35th ACM Symposium on Theory of Computing* (2003), pp. 59–68, arXiv:quant-ph/0209131.

[5] A. M. Childs, *Quantum information processing in continuous time*, Ph.D. thesis, Massachusetts Institute of Technology (2004).

[6] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, *Efficient quantum algorithms for simulating sparse Hamiltonians*, Communications in Mathematical Physics **270**, 359 (2007), arXiv:quant-ph/0508139.

[7] N. Wiebe, D. W. Berry, P. Høyer, and B. C. Sanders, *Higher order decompositions of ordered operator exponentials*, Journal of Physics A **43**, 065203 (2010), arXiv:0812.0562.

[8] A. Papageorgiou and C. Zhang, *On the efficiency of quantum algorithms for Hamiltonian simulation*, to appear in Quantum Information Processing (2012), arXiv:1005.1318.

[9] N. Zagury, A. Aragão, J. Casanova, and E. Solano, *Unitary expansion of the time evolution operator*, Physical Review A **82**, 042110 (2010), arXiv:1008.1985.

[10] A. M. Childs and R. Kothari, *Simulating sparse Hamiltonians with star decompositions*, in *Theory of Quantum Computation, Communication, and Cryptography* (Springer, 2011), vol. 6519 of *Lecture Notes in Computer Science*, pp. 94–103, arXiv:1003.3683.

[11] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, *Quantum computation by adiabatic evolution*, arXiv:quant-ph/0001106.

[12] E. Farhi, J. Goldstone, and S. Gutmann, *A quantum algorithm for the Hamiltonian NAND tree*, Theory of Computing **4**, 169 (2008), arXiv:quant-ph/0702144.

[13] A. W. Harrow, A. Hassidim, and S. Lloyd, *Quantum algorithm for linear systems of equations*, Physical Review Letters **103**, 150502 (2009), arXiv:0811.3171.

[14] J. T. Merrill and K. R. Brown, *Progress in compensating pulse sequences for quantum computation* (2012), arXiv:1203.6392.

[15] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and Quantum Computation* (AMS, 2002).

[16] M. Suzuki, *General theory of fractal path integrals with applications to many-body theories and statistical physics*, Journal of Mathematical Physics **32**, 400 (1991).

[17] D. Poulin, A. Qarry, R. Somma, and F. Verstraete, *Quantum simulation of time-dependent Hamiltonians and the convenient illusion of Hilbert space*, Physical Review Letters **106**, 170501 (2011), arXiv:1102.1360.

[18] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, vol. 55 of *Applied Mathematics Series* (U.S. Government Printing Office, 1964).

[19] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, *Strengths and weaknesses of quantum computing*, SIAM Journal on Computing **26**, 1510 (1997), quant-ph/9701001.

[20] E. Farhi and S. Gutmann, *Analog analogue of a digital quantum computation*, Physical Review A **57**, 2403 (1998), quant-ph/9612026.

[21] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, *Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms*, Journal of Magnetic Resonance **172**, 296 (2005).

[22] T. Schulte-Herbrüggen, A. Spörl, N. Khaneja, and S. J. Glaser, *Optimal control-based efficient synthesis of building blocks of quantum algorithms: A perspective from network complexity towards time complexity*, Physical Review A **72**, 042331 (2005), arXiv:quant-ph/0502104.

[23] T. W. Borneman, C. E. Granade, and D. G. Cory, *Parallel information transfer in a multi-node quantum information processor*, Physical Review Letters **108**, 140502 (2012), arXiv:1107.4333.

[24] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge U.K., 2000).

[25] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, et al., *Towards quantum chemistry on a quantum computer*, Nature Chemistry **2**, 106 (2009), arXiv:0905.0887.

[26] J. T. Barreiro, M. Müller, P. Schindler, D. Nigg, T. Monz, M. Chwalla, M. Hennrich, C. F. Roos, P. Zoller, and R. Blatt, *An open-system quantum simulator with trapped ions*, Nature **470**, 486 (2011), arXiv:1104.1146.

[27] S. Raeisi, N. Wiebe, and B. C. Sanders, *Quantum-circuit design for efficient simulations of many-body quantum dynamics* (2011), arXiv:1108.4318.

[28] A. Y. Kitaev, *Fault-tolerant quantum computation by anyons*, Ann. Phys. **303**, 2 (2006).

[29] L. Sheridan, D. Maslov, and M. Mosca, *Approximating fractional time quantum evolution*, Journal of Physics A **42**, 185302 (2009), arXiv:0810.3843.

[30] A. Y. Kitaev, *Quantum computations: Algorithms and error correction*, Russian Mathematical Surveys **52**, 1191 (1997).

[31] C. M. Dawson and M. A. Nielsen, *The Solovay-Kitaev algorithm*, Quantum Information and Computation **6**, 81 (2006), arXiv:quant-ph/0505030.

[32] V. Kliuchnikov, D. Maslov, and M. Mosca, *Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and T gates* (2012), arXiv:1206.5236.

[33] A. Bocharov and K. M. Svore, *A depth-optimal canonical form for single-qubit quantum circuits* (2012), arXiv:1206.3223.

[34] S. Blanes, F. Casas, and J. Ros, *Extrapolation of symplectic integrators*, Celestial Mechanics and Dynamical Astronomy **75**, 149 (1999).

[35] S. Chin, *Multi-product splitting and Runge-Kutta-Nyström integrators*, Celestial Mechanics and Dynamical Astronomy **106**, 391 (2010), arXiv:0809.0914.

[36] A. M. Childs and N. Wiebe, *Hamiltonian simulation using linear combinations of unitary operations*, Quantum Information and Computation **12**, 901 (2012), arXiv:1202.5822.